THE ART OF MATHEMATICAL PROGRAMMING
MY WORK WITH NELSON MACULAN

Leo Liberti
LIX, Ecole Polytechnique
91128 Palaiseau, France
liberti@lix.polytechnique.fr

Dedicated to Nelson Maculan on the occasion of his 70th birthday.

ABSTRACT

This paper is about my work with Nelson Maculan over a decade. It is a tour of mathematical programming applied to Nelson's favourite problems, in their natural chronological order: fundamental cycle bases in graphs (graph theory), the kissing number problem (combinatorial geometry), multiprocessor scheduling with communication delays (combinatorial optimization), the global solution of the Hartree-Fock equations (quantum chemistry), finding molecular conformations using NMR data (proteomics), configuration of the gamma-knife neurosurgery machine (medicine), and Nelson's pet problem, the Euclidean Steiner Tree.

KEYWORDS. Combinatorial optimization. Global optimization. Eucliean geometry.

Main area: Mathematical Programming.

1. Introduction

I met Nelson Maculan at the beginning of 2003, when I took up a postdoctoral fellowship at Politecnico di Milano, Italy. Nelson was completing a couple of years of visiting professorship in the Operations Research group at the department of electronics and information, which at that time was headed by F. Maffioli. I had recently finished a Ph.D. in Global Optimization (GO), and, as my luck had it, Nelson was just about the only researcher in that group who was actively working on nonlinear problems, as the rest of the group was mostly focused on Combinatorial Optimization (CO). Like everyone else, I found Nelson's spirits and enthusiasm irresistible.

Our overlap at Politecnico lasted about one year, but he nonetheless made an everlasting impression on me. During that year, we tried to find minimum fundamental cycle bases in graphs, we tested new Mathematical Programming (MP) formulations for packing equal spheres on the surface of another (equal) sphere in arbitrary dimensions, and started thinking about editing a book on global optimization methods, which eventually became a reality (Liberti and Maculan, 2006). Nelson had been invited to give a plenary talk at the Serbian OR society's national summer meeting (Sym-Op-Is), and brought me along with him. He introduced me to all his friends in Belgrade; and in particular to Tanja Davidovic, with whom we started collaborating on a scheduling problem that had nonlinearities in the constraints.

In 2004, he invited me to Rio de Janeiro for the first time, and sent me to replace him as plenary speaker at a "didactical conference" in Lima. Since he had little time for me in Rio, he told his ex-Ph.D. student Carlile Lavor to take good care of me. Carlile still remembers how he feared he would have to take care of the umpteenth old and boring "big prof" colleague of Nelson's, who would no doubt come up with plenty of bizarre touristic desires, all of them to be satisfied by himself. After I appeared, Carlile became ecstatic to find himself the senior of this new visitor of Nelson's. He sat me down at his wooden desk

and threw two beautiful problems at me, both of them cooperations with Nelson and other people. Carlile's secondary interest is bioinformatics, and, accordingly, the two new problems had to do with, respectively, quantum chemistry and proteomics. Both of them were global optimization problems with many nonlinear terms and no integer variables, so I felt I was on safe ground. I suggested MP formulations, reformulations and methods, and successfully tested some instances. I went back to Brazil again several times in order to continue my work with Nelson, Carlile and their colleagues, and several times Nelson and Carlile came to visit me in Milan and then Paris after I moved to Ecole Polytechnique in 2005.

At the end of 2007, Nelson wrote to me that Yue Zhang, a Chinese student of computer science who held an internship in the neurosurgery department of a Shanghai hospital, had written to him on account of a paper of ours on spherical packings and the kissing number problem (Kucherenko et al., 2007). Yue had tried to work out a formulation for his problem and failed, then asked for our help. This gave rise to a joint paper (Liberti et al., 2009b) on the configuration of the "gamma-knife" machine for performing non-invasive neurosurgery.

And finally, just a few days ago, Nelson came to see me in Paris and we talked about the Euclidean Steiner Tree problem, which is possibly his "pet problem". We dug up his formulations from 2000 and 2004, and tested them with today's solvers. We found to our astonishment that his first formulation, which he is really (and justly) proud of, performed beautifully with the IPOPT solver, which was able to find degenerate as well as nondegenerate Steiner trees in impressively short times.

This paper is an account of all the beautiful optimization problems that Nelson made me discover, and of some of the "art of mathematical programming" that Nelson created. I also took the opportunity to publish here a couple of minor unpublished results (for lack of time) obtained during the last few years.

2. The minimum fundamental cycle basis problem

It is well known that cycles in a graph $G=(V,E)$ form a vector space over the two-element field $\mathbb{F}_2 = \{0,1\}$. We map each cycle to its edge incidence vector: for example, the triangle {1,2,3} in the complete graph on 4 vertices is represented as follows:

{1,2}	{1,3}	{1,4}	{2,3}	{2,4}	{3,4}
1	1	0	1	0	0

i.e. with the 6-vector (1,1,0,1,0,0). The square given by the circuit (1,2,3,4) corresponds to the cycle vector (1,0,1,1,0,1). The boolean sum of these cycles is (0,1,1,1,1,0), i.e. the circuit (1,3,2,4). Of course, in this representation, cycles may well fail to be simple. Like any vector space, the cycle space has a basis, which allows one to express the exponential number of cycles in a graph by means of a polynomial subset thereof. The cost of a cycle is the cost of the edges that compose it, and the cost of a cycle basis is the sum of the costs of each cycle in the basis. Horton (1997) proved that one can find a minimum cost cycle basis in polynomial time. This is unlikely (unless **P**=**NP**) in the case where the basis is also required to be fundamental, i.e. there is a spanning tree T of G such that each cycle in the basis is the union of exactly one edge e in $E\setminus T$ and the edges of the unique path in T joining the endpoints of e.

Fundamental cycles bases (FCB) allow the compact representation of a whole cycle space by means of a spanning tree, and are useful in the classification of ring compounds, in the analysis of electrical circuits, and in other fields. We presented several MP formulations in Liberti et al. (2005), all of them based on binary decision variables set to 1 if a given vertex or edge belongs to a given cycle. These formulations only allowed us to solve very small instances to optimality. Part of our problem, in looking for better formulations, was to devise a polynomial number of variables or constraints for modelling a spanning tree.

Another part of the problem was to be able to sum tree edge weights as many times as they were part of a cycle. Nelson came up with a beautiful formulation, based on multicommodity flows, that turned out to be tighter, and to scale much better than our previous ones. Each commodity flow marks the unique (shortest) path in T between the endpoints of each edge e in E. If e is not in T, the flow defines the rest of the corresponding fundamental cycle. Let A be the set of arcs consisting of pairs of antiparallel arcs for each edge in E, and $G'=(V,E\cup A)$. For each edge $\{i,j\}$ in E, w_{ij} denotes the given edge weight, z_{ij} is a decision variable equal to 1 if and only if the edge is in T, and denotes the flow on the arc (i,j) in A corresponding to the commodity $\{k,l\}$ in E. The formulation below appeared in Amaldi et al. (2009).

$$
\begin{aligned}
\min \sum_{\{k,l\}\in E} \sum_{(i,j)\in A} w_{ij} x_{ij}^{kl} + \sum_{\{i,j\}\in E} w_{ij}(1-2z_{ij}) & \\
\forall \{k,l\} \in E \quad \sum_{j\in\delta(k)} (x_{kj}^{kl} - x_{jk}^{kl}) &= 1 \\
\forall \{k,l\} \in E,\ \forall i \in V\backslash\{k,l\} \quad \sum_{j\in\delta(i)} (x_{ij}^{kl} - x_{ji}^{kl}) &= 0 \\
\forall \{k,l\} \in E,\ \forall \{i,j\} \in E \qquad x_{ij}^{kl} &\leq z_{ij} \\
\forall \{k,l\} \in E,\ \forall \{i,j\} \in E \qquad x_{ji}^{kl} &\leq z_{ij} \\
\sum_{\{i,j\}\in E} z_{ij} &= n-1 \\
\forall \{k,l\} \in E,\ \forall (i,j) \in A \qquad x_{ij}^{kl} &\geq 0 \\
\forall \{i,j\} \in E \qquad z_{ij} &\in \{0,1\}.
\end{aligned}
$$

The first two constraints define multicommodity flows; the next two mark the relationship between tree variables z and flow variables x; the fifth constraint declares the cardinality of a spanning tree. The crucial insight is the objective function: this is what took Nelson the longest to get right. We were in his hotel room in Herceg-Novi, and I remember the multiple edits and computational trials before he finally managed it. Nelson argued (in his head) the following proof for expressing the cost $\Gamma(G)$ of the FCB of G:

$$
\begin{aligned}
\Gamma(G) &= \sum_{\substack{\{k,l\}\notin T \\ (i,j)\in A}} w_{ij} x_{ij}^{kl} + \sum_{\{k,l\}\notin T} w_{kl} \\
&= \Big(\sum_{\substack{\{k,l\}\in E \\ (i,j)\in A}} w_{ij} x_{ij}^{kl} - \sum_{\{k,l\}\in T} w_{kl}\Big) + \sum_{\{k,l\}\in E} w_{kl}(1-z_{kl}) \\
&= \sum_{\substack{\{k,l\}\in E \\ (i,j)\in A}} w_{ij} x_{ij}^{kl} - \sum_{\{k,l\}\in E} w_{kl} z_{kl} + \sum_{\{k,l\}\in E} w_{kl}(1-z_{kl}) \\
&= \sum_{\substack{\{k,l\}\in E \\ (i,j)\in A}} w_{ij} x_{ij}^{kl} + \sum_{\{k,l\}\in E} w_{kl}(1-2z_{kl}).
\end{aligned}
$$

The first line decomposes $\Gamma(G)$ in the cost of all the unique paths in T between endpoints of non-tree edges, plus the cost of the non-tree edges. The second line changes both terms. It exploits the fact that the sum of all flows between the endpoints of tree edges is simply the cost of T. Moreover, it moves T from quantifier to argument in the second term, using a standard technique. The rest are simple algebraic manipulations.

3. The kissing number problem

Given positive integers *D*, *N*, the decision version of the Kissing Number Problem (KNP) asks whether *N* unit (*D*-1)-spheres can be placed adjacent to a unit (*D*-1)-sphere centered in the origin. The optimization version asks for the maximum possible *N*. The pedigree of this problem is remarkable, having originated in a discussion between Isaac Newton and David Gregory. The name comes from billiard jargon: when two balls touch, they are said to "kiss". One may almost picture the two British chaps going down the pub, arm in arm, for a game of pool and a pint of ale; and then, in the fumes of alcohol, getting into a brawl about whether twelve or thirteen nonoverlapping spheres might kiss a central one if the table was three-dimensional. In Maculan et al. (1996), Nelson proposed a Mixed-Integer Nonlinear Programming (MINLP) formulation for solving the optimization version of the KNP. This being 1996, Nelson had no access to a solid, general-purpose MINLP solver, and hence set out to describe a Lagrangian relaxation approach, which was eventually tested by Pietro Belotti several years later with disappointing results. To me, Nelson only gave the basic NLP formulation for the decision version:

$$\left.\begin{array}{rrcl} \max\limits_{\alpha\geq 0, x\in\mathbb{R}^D} & \alpha & & \\ \forall i \leq N & \|x_i\|^2 & = & 4 \\ \forall i < j \leq N & \|x_i - x_j\|^2 & \geq & 4\alpha \end{array}\right\} \quad \text{[KNP]}$$

The main decision variables are the centers x_i of the *N* unit spheres kissing the central unit sphere. The first constraints state that the spheres must touch the central sphere exactly, and the second constraints require the interior of the spheres to be pairwise nonoverlapping unless $\alpha < 1$. The α tolerance in the nonoverlapping constraints serves to avoid numerical issues with most local NLP solvers. Nelson's intentions were to use a spatial BB (sBB) algorithm I had coded during my Ph.D. to solve the smallest open case (*D*=4), which was uncertain between *N*=24 and *N*=25. Unfortunately, my sBB was too inefficient for this instance: we let it run for weeks without making much progress. In Kucherenko et al. (2007), we introduced a simple reformulation of [KNP] which allowed two stochastic GO solvers, one based on Multi-Start (MS) and the other on Variable Neighbourhood Search (VNS), to give strong indications that the correct *N* is 24. Just as we were finishing our computational results, O. Musin published a formal proof of this fact.

4. Multiprocessor scheduling with communication delays

The focus of the Ph.D. completed by our friend Tanja Davidovic at the Mathematical Institute of the Serbian Academy of Sciences and Arts in Belgrade was a special type of scheduling problem arising from compiler design in a parallel computation setting. The Multiprocessor Scheduling Problem with Communication Delays (MSPCD) consists in finding a minimum makespan schedule to run a set *V* of tasks with fixed lengths on an arbitrary network defined on a set *P* of homogeneous processors. The task precedence is modelled by a directed acyclic graph (DAG) on the set of tasks whose arcs represent task precedence. If two tasks with a precedence relation are executed on different processors, they incur a communication cost depending both on the distance between the processors in the network topology, and on the amount of exchanged data between the tasks. The last condition inevitably leads to multiplying decision variables controlling distance in the processor network topology and exchanged data. Perhaps because of this, the MSPCD had never been formulated in full generality before: only partial subproblems (such as the assignment part of the scheduling) and special cases (e.g. where the processor network was a complete graph) had previously appeared in the literature. In Davidovic et al. (2007), we introduced two formulations for this new problem. The first is based on a standard extension of a classic scheduling MP formulation. The second is more innovative, and is based on a modification of rectangle packing. Both were linearized exactly using

standard and nonstandard methods, and yielded Mixed-Integer Linear Programming (MILP) formulations that were fed to CPLEX in order to compute the guaranteed global optimum. It turns out that the packing formulation performs much better than the classic one.

5. The Hartree-Fock problem

This is the first of two problems that were introduced to me by Nelson's ex-Ph.D. student Carlile Lavor, now associate professor at the University of Campinas (Carlile's career and my own match quite closely: we became assistant, then associate professors, then department vice-presidents, more or less concurrently). Carlile had, since the beginning of his collaboration with Nelson, looked specifically for bioinformatics applications of optimization techniques. One day, Nelson told him to visit his quantum chemist colleague M.A. Chaer Nascimento, who then described the Hartree-Fock (HF) method to Carlile. This computational technique arises in quantum chemistry in order to compute approximated solutions of the time-independent non-relativistic Schrödinger equation $H\Phi = E\Phi$, where H is the Hamiltonian of a system of nuclei and electrons in a molecule, E is the energy of the system, and Φ is the solution of the equation. For closed-shell systems, and limited to the spatial position of the electrons in the molecule, Φ consists in a set $\{\varphi_i\}$ of n spatial orbital functions. These functions can only be computed analytically for the simplest cases, leaving computational approximation as the only viable alternative. Each φ_i is expressed as a linear combination

$$\phi_i = \sum_{r \leq b} c_{ri} \chi_r$$

of a finite number of basis functions χ_r. The approximation becomes exact in the limit for $b \to \infty$, in which case $\{\chi_r\}$ is a basis of the Hilbert space. The problem input consist in all the inner products $S_{rs} = \langle \chi_r, \chi_s \rangle$ and in other constants used to express the energy in terms of the unknown coefficients c_{ri}. Approximations are best whenever the energy value of the system is lowest. For closed-shell systems, the φ_i are known to be orthonormal, so that $\langle \varphi_i, \varphi_j \rangle = \delta_{ij}$ for all $i \leq j$ (the Kronecker delta symbol which has value 1 if and only if $i=j$ and 0 otherwise). For open-shell systems, the φ_i are only required to be linearly independent. The HF method is roughly equivalent to a local descent in the c space. As such, it suffers from two drawbacks: the need for and instability with respect to an initial solution, and the fact that the descent might terminate in a local, rather than global, optimum. Carlile and Chaer had published a first work (Lavor et al. 2005) where a guaranteed global optimization method (interval BB) was employed on the following formulation:

$$\left.\begin{array}{lrcl} \min\limits_{c \in \mathbb{R}^{|B||N|}} & E^{\text{closed}}(c) & & \\ \forall i \leq j \in N & \sum\limits_{r,s \leq b} S_{rs} c_{ri} c_{sj} & = & \delta_{ij} \\ & c^L \leq c & \leq & c^U, \end{array}\right\} \quad \text{[HFP]}$$

where the objective function E^{closed} is a quartic polynomial in the c variables, and the variable bounds c^L, c^U are given. In the joint works Lavor et al. (2007) and Liberti et al. (2009a), one for the physics and the other for the OR community, Carlile, Chaer, Nelson and I solved [HFP] and a reformulation thereof (based on replacing the bilinear products $c_{ri}c_{sj}$ by a linearization variable w_{rsij}) using: (i) the sBB implementation I had coded during my Ph.D.; and (ii) and two metaheuristics (the same MS and VNS that had been used for [KNP]). We were able to compute optimal approximations for the helium (He) and beryllium (Be) atomic systems. In 2009 we used VNS to also compute approximations for the neon (Ne) and magnesium (Mg) systems, but these results remained unpublished. We report them here for the first time. In the table below, V and C denote the number of variables and

constraints in the [HFP] formulation.

Atom	V	C	E (au)	CPU (s)
He	2	1	-2.747064059541913	0.02
Be	4	3	-14.351912029941255	0.04
Ne	25	15	-126.642714605022974	0.48
Mg	54	21	-196.758558903276736	19.87

We remark that our input data has been obtained in a simplified way by our quantum chemist co-authors; accordingly, the values for *E* might have a small difference with respect to the known values.

Ongoing work focuses on the adaptation of [HFP] to open-shell systems, which calls for constraints on *c* enforcing linear independence. Although a famous MP relaxation of linear independence yields the well-known Semidefinite Programming (SDP) paradigm, the original requirement poses a considerable challenge in MP, as emphasized in Liberti (2012).

6. The molecular distance geometry problem

The Molecular Distance Geometry Problem (MDGP) is the second problem proposed to me by Carlile. The MDGP consists in finding a realization $x : V \to \mathbb{R}^3$ of a given weighted simple undirected graph $G=(V,E,d)$ where d_{uv} is the weight of the edge $\{u,v\}$ in E, so that the following is satisfied:

$$\forall \{u, v\} \in E \quad \|x_u - x_v\|_2 = d_{uv}.$$

The above formulation is a system of nonlinear equations. A few tests show that only the smallest graphs can be realized by solving it computationally. Reformulating it to the following penalty minimization problem:

$$\min_{x \in \mathbb{R}^{3n}} \sum_{\{u,v\} \in E} (\|x_u - x_v\|_2^2 - d_{uv}^2)^2$$

yields a nonconvex NLP formulation, which was tested with varying degrees of success in Lavor et al. (2006a).

The generalization of the MDGP to arbitrary dimensions (i.e. x maps V into a Euclidean space of arbitrary dimension K) is called the Distance Geometry Problem (DGP), and has applications to clock synchronization (K=1), wireless sensor networks (K=2), proteomics, robotics and statics (K=3) and data visualization (arbitrary K). We studied it from the point of view of an application to bioinformatics: the computation of the 3D structure of a protein backbone given a set of inter-atomic distances computed using Nuclear Magnetic Resonance (NMR) experiments. The MDGP is the inverse problem of computing the atomic positions given the distances.

In 2005, Carlile discovered that the input data G relative to protein backbones have a special structure, i.e. there exists an order on $V=\{1, ..., n\}$ such that: (i) x_v is known for each v in $\{1, ..., K\}$, and (ii) the K immediate predecessors of each $v > K$ are all adjacent to v in G. This implies that x_v is at the intersection of the K spheres centered at x_{v-i} with radius $d_{v-i,v}$ for each i in $\{1, ..., K\}$. Since, in general, the intersection of K spheres has two points, there are at most two feasible positions for x_v once its predecessors have been placed. Moreover, these two positions must also be feasible with all distances from v to all its adjacent predecessors (not just the immediate ones). This suggests a very natural recursive algorithm where, at each level, the search is recursed in each of the (at most) two possible positions. We explored the theoretical and empirical properties of this algorithm, called Branch-and-Prune (BP), along many directions in a long (and continuing) sequence of papers started with Lavor et al. (2006b). See Lavor et al. (2012) for a recent survey, co-

authored with Antonio Mucherino, who, since 2009, became an essential and indispensable pillar of the MDGP team.

The presence of an order with the given properties defines a subset of DGP instances that we called ^{K}DMDGP. This problem turns out to be **NP**-hard, and no MP formulation is known. Its restriction to K=3 is called DMDGP. We give here a new formulation for the DMDGP.

If the three immediate predecessors of each vertex $v > 3$ are also adjacent to v, then the graph consists of a set of n-3 successive 4-cliques, plus perhaps some other edges. Each 4-clique, when realized in the three-dimensional Euclidean space, defines four points $\{x_{v-3}, x_{v-2}, x_{v-1}, x_v\}$, which we can assume to be distinct with probability 1. We denote P_v and Q_v the two planes passing through x_{v-3}, x_{v-2}, x_{v-1} and x_{v-2}, x_{v-1}, x_v in turn. The inner product of the normal vectors to P_v and Q_v is proportional to the dihedral angle θ_v between the two planes. It is not hard to show that, given a DMDGP instance, all dihedral angles θ_v for $v > 3$ can be computed in constant time. For each $v > 2$, we denote by γ_v the normal vector to x_{v-2}, x_{v-1}, x_v. The corresponding MP formulation has decision variables x_{vk} for all $v > 3$ and $k \leq 3$:

$$\begin{aligned}
\min_{\substack{x\in\mathbb{R}^{3n}\\ \gamma\in\mathbb{R}^{3(n-3)}}} \quad & \sum_{\{u,v\}\in E} (\|x_u - x_v\|^2 \;-\; d_{uv}^2)^2 \\
\forall v > 3 \quad & \sum_{k\leq 3} \gamma_{v-1,k}\gamma_{vk} \;=\; \|\gamma_{v-1}\|\|\gamma_v\|\cos\theta_v \\
\forall v > 3 \quad & (x_{v-2,2} - x_{v-1,2})(x_{v3} - x_{v-1,3}) - \\
& -(x_{v-2,3} - x_{v-1,3})(x_{v2} - x_{v-1,2}) \;=\; \gamma_{v1} \\
\forall v > 3 \quad & (x_{v-2,1} - x_{v-1,1})(x_{v3} - x_{v-1,3}) - \\
& -(x_{v-2,3} - x_{v-1,3})(x_{v1} - x_{v-1,1}) \;=\; \gamma_{v2} \\
\forall v > 3 \quad & (x_{v-2,1} - x_{v-1,1})(x_{v2} - x_{v-1,2}) - \\
& -(x_{v-2,2} - x_{v-1,2})(x_{v1} - x_{v-1,1}) \;=\; \gamma_{v3}
\end{aligned}$$

where the first constraints relate the inner products of the γ_v with the dihedral angles, and the last three constraints relate the γ_v with the x variables. We remark that γ_3 can be computed from the given values for x_1, x_2, x_3. Although this formulation can be extended to different values of K by adapting the relations between γ_v and x, these relations are given by a determinant, and we are not aware of any explicit non-recursive formula for computing the determinant of square matrices of non-fixed size. The above nonconvex NLP formulation was never tested computationally.

7. Configuration of gamma-knife neurosurgery units

This is an example of a practical problem whose exact formalization is difficult or perhaps impossible. Some forms of brain tumors can be cured non-invasively by intersecting several gamma rays into a tumor spot. This only kills the tumor cells in a sphere centered at the intersection point, and spares those outside the sphere. The sphere radii depend linearly on the width of the gamma rays and monotonically nonlinearly on the time of irradiation. The aim is to kill all tumor cells, as even leaving just a few alive makes the tumor grow back, and spare all healthy cells (accidentally killing healthy brain cells might cause severe damage). Since tumors have various shapes and sizes, and having a shape that can be covered by a set of spheres exactly is practically impossible, the problem has no solution. Several rules-of-thumb approximations in both the coverin and the irradiation extent are used instead, but none of them offers a definitive cure or a guarantee to spare all healthy cells.

Existing literature pre-computes the irradiation intensity at a point x in function of the distance to all irradiation centers. Since these are unknown, space is treated as a

Euclidean 3D grid (whose elements are called voxels), and the intensity is computed for every possible assignment of a center to a voxel. Since the intensity is additive, it suffices to compute it for one irradiation center at a time. MILP formulations are then used to decide the assignments of irradiation centers to voxels and the irradiation time per center so that various heuristic objective functions are optimized: e.g. total irradiation, irradiation of healthy cells, and so on. The constraints are usually enforced on the total irradiation inside the tumour, which must be sufficient to kill it, and outside of it (this must never exceed a certain safety threshold). Since space is discretized, these constraints can never guarantee anything along the border between the tumour and the healthy tissue, unless the grid is very fine. But if this is the case, then the amount of input data is enormous and the resulting MILP cannot be solved.

Our approach consisted in trying to replace the precomputed data by the output of a sphere covering subproblem which gives the optimal choice for assigning irradiation centers to points in space. The "exact" form of this problem is an infinite formulation with sums replaced by integrals, which we do not know how to solve. In Liberti et al. (2009b) we propose a set of finite approximations to this infinite problem, which we solve using MINLP and MILP solution techniques, with varying degrees of success. Ongoing work focuses on integrating this subproblem to the problem that decides the irradiation time optimally.

8. The Euclidean Steiner Tree problem

The Euclidean Steiner Tree Problem (ESTP) is as follows. Given a set $N = \{p_1, ..., p_n\}$ of points in a K-dimensional Euclidean space, find a finite set S of points (not necessarily distinct from those in N) and a spanning tree T in the complete graph on $V = N \cup S$ such that the cost of the tree, computed as the sum of the Euclidean length of the edges in the tree, is minimum.

This very general setting surprisingly has a very rich mathematical structure. The following facts are known to hold at optima of the ESTP for any integer $K > 1$: (a) $|S| = n$-2; (b) vertices in N can only be adjacent to vertices in S; (c) vertices in N necessarily have stars of size 1; (d) vertices in S necessarily have stars of size 3; (e) each vertex in S is coplanar with its three adjacent vertices; (f) if $q \in S$ and r, s are distinct points adjacent to q in an optimal Steiner tree, then the vectors r-q, s-q must form a plane angle of $2\pi/3$ radians. Any valid MP formulation of the ESTP must include continuous decision variables x_v for points in S, binary decision variables y_{uv} to decide whether there is an edge between two vertices u and v, and a Euclidean (hence nonlinear) cost function on edges. This necessarily yields a MINLP. Properties (a)-(f), if expressed as constraints, provide valid inequalities. Nelson, together with Ph. Michelon and A. Xavier, proposed a MINLP formulation in Maculan et al. (2000), which I find a masterwork of deceptive simplicity, and which, moreover, works well in practice.

We replace N, S by sets of indices indexing points: $N = \{1, ..., n\}$ and $S = \{n+1, ..., n+m\}$. We employ decision variables $x_v \in \mathbb{R}^K$ for all v in V (and fix $x_v = p_v$ for all v in N), as well as binary variables y_{uv} set to 1 if and only if $\{u,v\}$ is an edge in the Steiner tree T. Requirement (a) above is easily enforced by setting $m = n$-2. We consider a set of potential edges E on V defined as follows:

$$E = \left(\bigcup_{u \in N} \{\{u, v\} \mid v \in S\} \right) \cup \{\{v, z\} \mid v, z \in S\}.$$

In other words, E is the union of all the stars in S centered around vertices in N, as well as the clique in S. This definition of E, which neglects all edges involving two vertices in N, enforces requirement (b) above. Since T is a spanning tree for $G = (V,E)$, then the induced subtree $T[S]$ of S must be a spanning tree for $G[S]$, i.e. the clique on S. By requirement (c),

extending $T[S]$ to T is easy, since each vertex in N is pendent in T: it suffices to enforce the constraints:

$$\forall u \in N \quad \sum_{v \in S} y_{uv} = 1.$$

Requirement (d) is also easy to enforce:

$$\forall v \in S \quad \sum_{u \in N} y_{uv} + \sum_{\substack{u \in S \\ u < v}} y_{uv} + \sum_{\substack{u \in S \\ u > v}} y_{vu} = 3$$

Requirements (e)-(f) are of a more geometrical nature, and constrain the x variables rather than the tree topology encoded in the y variables. Accordingly, we do not explicitly encode them as constraints in the formulation (optimal Steiner trees are supposed to satisfy them anyway).

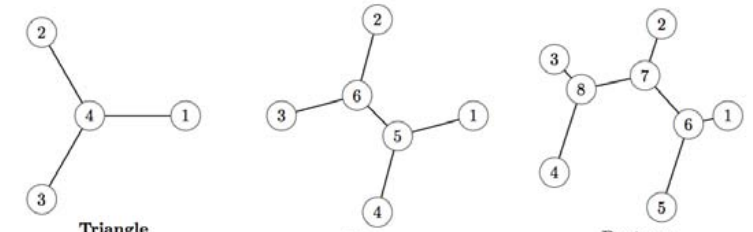
The only conditions left to enforce is that y, restricted to vertices in S, should define a spanning tree on the clique on S. According to Nelson, this is the point where he spent most time and effort. And this is, still according to him, his most prized result in the art of MP modelling:

$$\forall v \in S \setminus \{n+1\} \quad \sum_{\substack{u \in S \\ u < v}} y_{uv} = 1;$$

his own explanation for these constraints is terse at best: in Maculan et al. (2000), they are said to "avoid the formation of subtours among the Steiner vertices". If we take a closer look, n+2 is forced to be adjacent to n+1, n+3 is forced to be adjacent to either n+2 or n+1, n+4 is forced to be adjacent to exactly one vertex in {n+3, n+2, n+1}, and so on. These constraints would, for example, forbid all trees not including the edge e_1 = {n+1, n+2}. Nelson, however, argues that this limitation is irrelevant, insofar as the geometrical position of the vertices in S is not given *a priori*, but decided by the decision variables x_v (for $v \in S$). The statement "T never contains e_1" is always false: it suffices that the positions x_{n+1} and x_{n+2} be assigned to points in such a way that e_1 will *necessarily* be in T. This holds similarly for all other topology limitations imposed on T by Nelson's constraints.

Nelson also remarks that whilst most spanning tree formulations are for labelled graphs, his "spanning tree formulation" is really midway between labelled and unlabelled graphs. He regrets that some topologically equivalent trees are allowed by his formulation, and leaves the MP formulation defining all unlabelled spanning trees in a clique and, alternatively, in a given graph, as open problems in the art of mathematical programming.

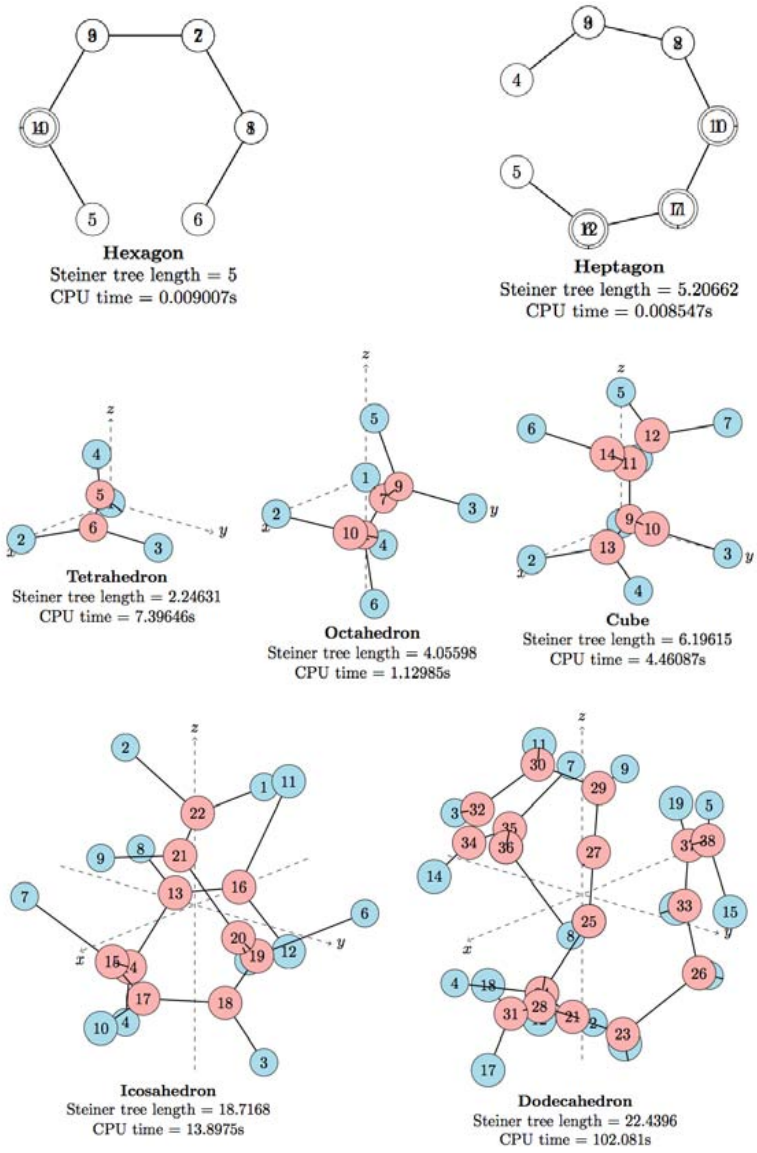
Below, we list some optimal Steiner trees for $K \in \{2,3\}$ found using Nelson's formulation and today's solver technology, i.e. the local NLP solvers IPOPT and SNOPT used within a MS metaheuristic. We remark that IPOPT does not necessarily provide integral values for integer variables: the fact that optimal Steiner trees were found (with optimal binary values for the y variables) using this methodology is a very strong point in favour of Nelson's formulation.



Triangle
Steiner tree length = 3, CPU time = 0.010202s

Square
Steiner tree length = 3.8637
CPU time = 0.003692s

Pentagon
Steiner tree length = 4.57433
CPU time = 0.024557s



We remark that Nelson's formulation is just as accurate in degenerate instances, such as the hexagon and the heptagon, as in non-degenerate ones (all the others). Instances are degenerate when all the optimal Steiner trees have one or more points in S that are coincident with given points in V.

9. Conclusion and acknowledgments

This paper is a recollection of all the work that Nelson Maculan and I did together on the subject of mathematical programming and its applications. I am grateful to him as a teacher and as a friend, and to all our co-authors for helping us make all this work possible.

References

Amaldi, E., Liberti, L., Maffioli, F., Maculan, N. (2009). Edge-swapping algorithms for the minimum fundamental cycle basis problem, *Mathematical Methods of Operations Research*, **69**:205-223.

Davidovic, T., Liberti, L., Maculan, N., Mladenovic, N. (2007). Towards the optimal solution of the multiprocessor scheduling problem with communication delays, in *Proceedings of the MISTA Conference*, Paris.

Horton, J. (1987), A polynomial-time algorithm to find the shortest cycle basis of a graph, *SIAM Journal of Computing*, **16**(2):358-366.

Kucherenko, S., Belotti, P., Liberti, L., Maculan, N. (2007), New formulations for the Kissing Number problem, *Discrete Applied Mathematics*, **155**(14):1837-1841.

Lavor, C., Cardozo, T., Chaer Nascimento, M. (2005). Using an interval branch-and-bound algorithm in the Hartree-Fock method, *International Journal of Quantum Chemistry*, **103**:500-504.

Lavor, C., Liberti, L., Maculan, N. (2006a). Computational experience with the molecular distance geometry problem, in Pinter, J. (ed.), *Global optimization: Scientific and engineering case studies*, Springer, Berlin.

Lavor, C., Liberti, L., Maculan, N. (2006b). The discretizable molecular distance geometry problem, *arXiv* technical report `q-bio/0608012`.

Lavor, C., Liberti, L., Maculan, N., Chaer Nascimento, M. (2007). Solving Hartree-Fock systems with global optimization methods, *Europhysics Letters*, **77**(5):50006p1-p5.

Lavor, C., Liberti, L., Maculan, N., Mucherino, A. (2012). Recent advances on the discretizable molecular distance geometry problem, *European Journal of Operational Research*, **219**:698-706.

Liberti, L., Amaldi, E., Maculan, N., Maffioli, F. (2005), Mathematical models and a constructive heuristic for finding minimum fundamental cycle bases, *Yugoslav Journal of Operations Research,* **15**(1):15-24.

Liberti, L., Maculan, N., eds. (2006), *Global Optimization: from Theory to Implementation*, Springer, Berlin.

Liberti, L., Lavor, C., Chaer Nascimento, M., Maculan, N. (2009a). Reformulation in mathematical programming: an application to quantum chemistry, *Discrete Applied Mathematics*, **157**:1309-1318.

Liberti, L., Maculan, N., Zhang, Y. (2009b). Optimal configuration of gamma ray machine radiosurgery units: the sphere covering subproblem, *Optimization Letters*, **3**:109-121.

Liberti, L. (2012). Modelling rank constraints in mathematical programming, in S. Pickl et al. (eds.), *Proceedings of the 11th Cologne-Twente Workshop on Graphs and Combinatorial Optimization*, München.

Maculan, N., Michelon, Ph., MacGregor Smith, J. (1996). Bounds on the Kissing Numbers in $\mathbb{R}^n$: Mathematical programming formulations. Technical report. University of Massachusetts, Amherst, USA.

Maculan, N., Michelon, Ph., Xavier, A. (2000). The Euclidean Steiner tree problem in R^n: A mathematical programming formulation, *Annals of Operations Research*, **96**:209-220.